Blue Gene Consortium Day

29th March 2006

Project Title: Lattice Boltzmann for Complex Fluids

Document Title: Blue Gene Consortium Day: 29th March 2006

Document Identifier: EPCC-CD1 1.0

Document history:

Personnel	Date	Summary	Version
Kevin Stratford	April 2006	First Draft	1.0

Distribution List: *EPCC, IBM*

Contents

1 Introduction

In this report I will summarise the results of a number of different benchmarks performed on the Blue Gene system at IBM Thomas J. Watson Research Center as part of the Blue Gene "Consortium Day" on 29th March 2006.

The benchmarks represent system of the type we are currently investigating for physical properties, and so represent a fair test of the machine. Practical time constraints meant that I did not queue any runs larger than a 1024 cubed lattice (equal in size to the largest run I am aware of in the past) on more than 8 racks.

2 The Lattice Boltzmann Code

The code, *Ludwig* (named for Ludwig Boltzmann, 1844–1906), provides a numerical solution of the Navier-Stokes equations for fluid flow via the lattice Boltzmann method [1, 2]. In common with other numerical methods, it represents the flow domain as a discrete lattice. However, it also uses a discrete velocity space $\mathbf{c}_i$, where the $\mathbf{c}_i$ are chosen so that $\mathbf{c}_i \Delta t = \mathbf{r}_i$, where $\mathbf{r}_i$ are lattice vectors and Δt is the discrete time step.

The fluid is represented by a distribution function f_i, which can be thought of as the density of fictitious fluid particles at a given position having velocity $\mathbf{c}_i$. Physical quantities are defined in terms of this distribution function, for example, the density $\rho = \sum_{i=1}^{n} f_i$.

The time evolution is via a discrete Boltzmann equation, which can be thought of in two parts: first, a local "collision" stage, where appropriate physics is introduced at each lattice site to update the local distributions and second, a "propagation" stage in which each distribution moves $\mathbf{c}_i \Delta t$, i.e., one lattice site in the appropriate direction.

The discrete velocity set $\mathbf{c}_i$ is denoted DdQn, where d is the number of dimensions and n is the number of discrete velocities (usually 15 or 19 in three dimensions, connecting a given lattice site to its nearest neighbours, and next-nearest neighbours). The method therefore has a relatively high memory requirement (15 or 19 double precision state variables per lattice site for a single phase fluid). For a binary fluid, a second distribution is introduced, doubling the memory requirement.

The method has a number of advantages: the pressure calculation is entirely local, meaning that parallelisation does not require anything more than halo swaps between adjacent domains to accommodate the propagation stage; complex topological structure evolves naturally on the lattice in the case of binary mixtures, removing the need for complex interface tracking procedures. Finally, solid fluid boundary conditions for moving objects may be introduced [3] with little loss of efficiency [4].

Computationally, the collision stage is dominated by floating point operations, while in contrast the propagation stage has no floating point operations but is dominated by memory movement. (Lattice Boltzmann is unlikely to win any prizes for overall Flop rate.) Colloidal particles are based around linked list manipulation with a significant number of irregular memory accesses.

[0] Copy prepared: April 3, 2006 @ 12:24

3 Benchmarks

3.1 Description

I have run three different benchmarks:

1. **T1**: A binary fluid system on a $512 \times 1024 \times 512$ lattice undergoing spinodal decomposition. The binary fluid model has two distributions using D3Q19.
2. **T2**: A single fluid system on a 1024 cubed lattice with fluctuating hydrodynamics [5]. This uses D3Q15 and the problem is large enough that it can just be accommodated on a single rack.
3. **T3**: As for **T2**, but with the addition of colloidal particles with radius 4.77 lattice units at 25% volume fraction (579,465 particles in total).

These are realistic benchmarks which are indicative of the type of systems we are currently interested in investigating for a number of different physical problems.

Compiler options in all cases were

```
-O3 -qipa -qhot -qarch=440 -qtune=440
```

All the benchmarks use virtual node mode and have been run on between 1 and 8 racks (i.e., 2048 and 16384 MPI processes). Results are presented in Figure 1.

3.2 Results

The results for the smallest problem **T1** are very good, scaling almost perfectly to 8 racks. This is likely to be about the limit of scaling for this problem.

The performance of the **T2** problem is slightly puzzling compared with what I expected from the Edinburgh system. The times show a large contribution in the lattice halos, which also captures load imbalance. It appears that the fluctuations (essentially random number generation) have introduced some load imbalance in the problem which could be investigated further. However, the scaling remains robust, particularly as the local domain size is reduced.

In contrast, the **T3** run which is the same as **T2** except for the inclusion of colloidal particles. The inclusion of particles actually eliminates the problems associated with the lattice halos seen in **T2**. The computation time required for the particles is somewhat offset by a saving in the collision stage associated with non-fluid sites. Scalability here, although good to 8 racks, will probably be ultimately limited by extra communication costs associated with particles.

4 Summary

The lattice Boltzmann benchmarks run on the Watson system show good scaling to 8 racks compared with a single rack.

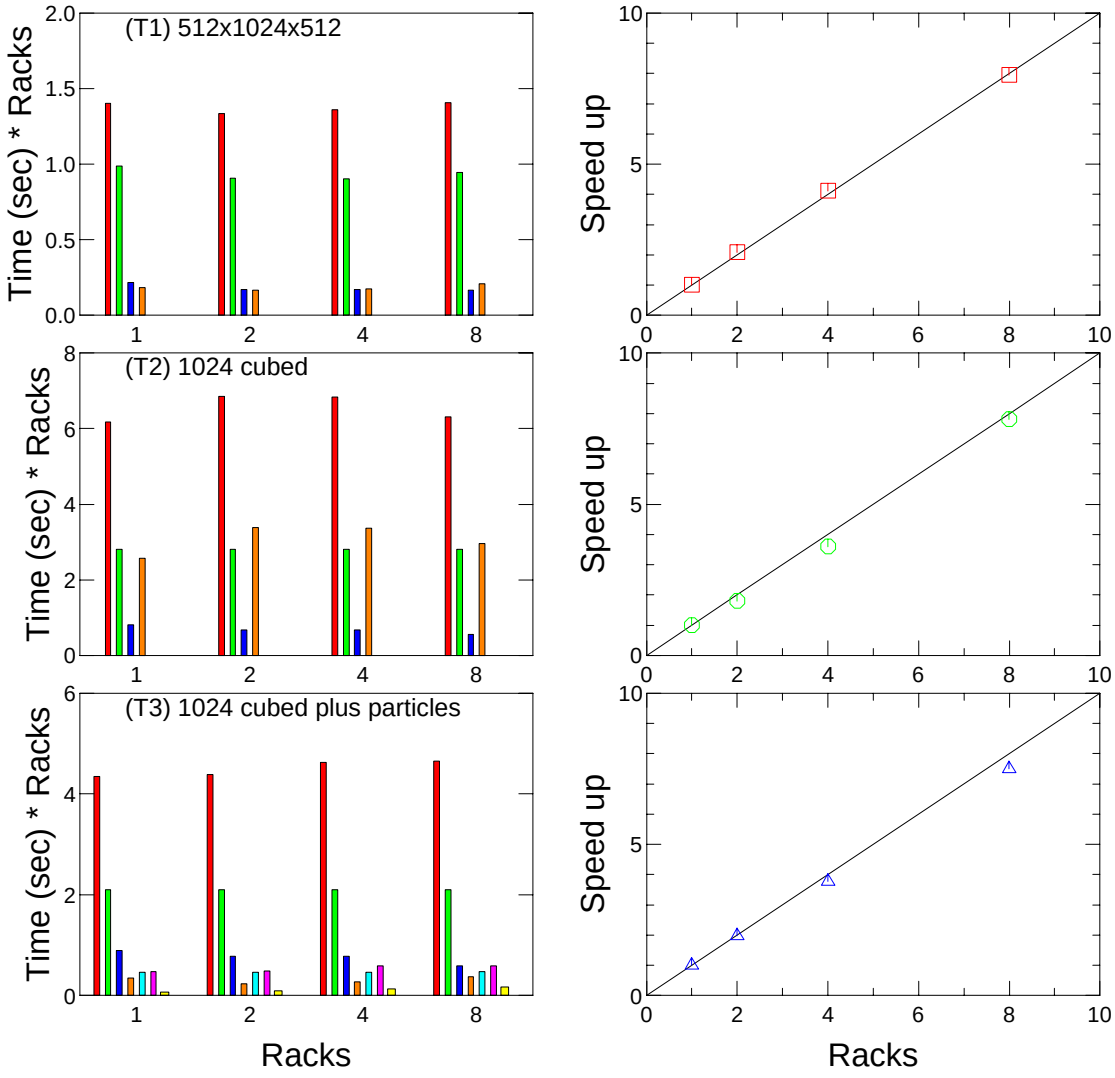

Figure 1: The breakdown of the time (left) and the scaling compared to one rack (right) as a function of the number of racks. The top panels show problem **T1**, the centre **T2**, and the bottom **T3**. The bars in the breakdown for each run are, from left to right, total, collision, propagation, lattice halos, (and in case **T3**) particle construction, particle boundary conditions, and particle halos. All the benchmarks use virtual node mode (i.e., 1 rack is 2048 MPI processes, while 8 racks is 16384 MPI processes).

Acknowledgements

Many thanks to Peter Mayes for hosting me at Bedfont Lakes on 29th, and for Jonathan Follows of IBM UK for setting this up. Support in the US was provided by Maria Eleftheriou and Fred Mintzer of IBM.

References

[1] S. Succi, *The lattice Boltzmann equation and beyond*, 2001.

[2] Desplat, J.-C., I. Pagonabarraga, and P. Bladon, LUDWIG: A parallel lattice-Boltzmann code for complex fluids. *Comput. Phys. Comms.*, **134**, 273, 2001.

[3] A.J.C. Ladd *J. Fluid Mech.*, **271**, 285, 1994; *J. Fluid Mech.*, **271**, 311, 1994.

[4] K. Stratford, and I. Pagonabarraga, Parallel domain decomposition for lattice Boltzmann with moving particles, preprint (2005).

[5] R. Adhikari, K. Stratford, M.E. Cates, and A.J. Wagner, Fluctuating Lattice Boltzmann, *Europhysics Letters*, **71**, 473 (2005).